1. Understanding Lookup Activity in Azure Data Factory

\* **Research what a Lookup activity is in Azure Data Factory (ADF).**

A **Lookup activity** in Azure Data Factory is used to **fetch data from a source** and **pass it to other activities** in a pipeline. Think of it like a **query tool** that **retrieves one or more rows** from a database, file, or API, and then **uses that data in the pipeline**.

\* **Explain its purpose and where it is commonly used in data pipelines.**

**Retrieve Configuration Data:** Get parameter values (e.g., file paths, table names) from a database.

**Check for Records:** Find if data exists before proceeding.

**Pass Data to Other Activities:** Use lookup results in **ForEach loops**, **If Condition**, or **Set Variable** activities.

**\* Provide examples of scenarios where a Lookup activity would be beneficial.**

**Scenario: Copy New Files Only**

* Use Lookup to query a database for the **last copied file date**.
* Pass the date to the **Copy Activity** to get only **new files**.

**Scenario: Conditional Data Processing**

* Lookup customer orders from a SQL table.
* If no orders exist (If Condition), **skip processing**.
* If orders exist, **proceed to process them**.

**Scenario: Dynamic Pipeline Configuration**

* Store **file paths and target tables** in a configuration table.
* Use Lookup to **read configurations**.
* Pass values to **ForEach activity** to process multiple tables/files.

2. Integration Runtimes in Azure Data Factory

**\* Define Integration Runtimes (IR) and their role in data movement**.

An **Integration Runtime (IR)** is the **compute infrastructure** used by Azure Data Factory to **move, transform, and integrate data** between different systems. Think of it as the **engine** that **executes activities** in your pipelines.

\* Identify and explain the different types of Integration Runtimes.

\* Describe when to use each type with real-world examples.

**Azure Integration Runtime (Cloud IR)**

* **What it is:** Runs activities in the **cloud** and connects to **Azure services** or **public endpoints**.
* **Use it when:**  
  ✔ Copying data between **Azure Blob Storage and Azure SQL**  
  ✔ Running **Data Flows** for transformations  
  ✔ Connecting to **Software as a Service (SaaS)** sources

**💡 Example:** Copying sales data from **Azure Data Lake** to **Azure SQL Database** for reporting.

**Self-Hosted Integration Runtime (Self-hosted IR)**

* **What it is:** Installed **on-premises** or on a **VM** to connect to **local data sources**.
* **Use it when:**  
  ✔ Moving data from **on-premises SQL Server** to the cloud  
  ✔ Accessing data behind a **firewall or private network**  
  ✔ Integrating with **legacy systems**

**💡 Example:** Migrating customer records from an **on-premises SQL Server** to **Azure Data Lake**.

**Azure-SSIS Integration Runtime**

* **What it is:** Runs **SQL Server Integration Services (SSIS)** packages in the **cloud**.
* **Use it when:**  
  ✔ You need to lift-and-shift existing **SSIS packages**  
  ✔ Running complex **ETL workflows**  
  ✔ Integrating with **SQL Server and SQL Agent Jobs**

**💡 Example:** Moving existing **SSIS packages** from **on-premises** to **Azure Data Factory** without rewriting them.

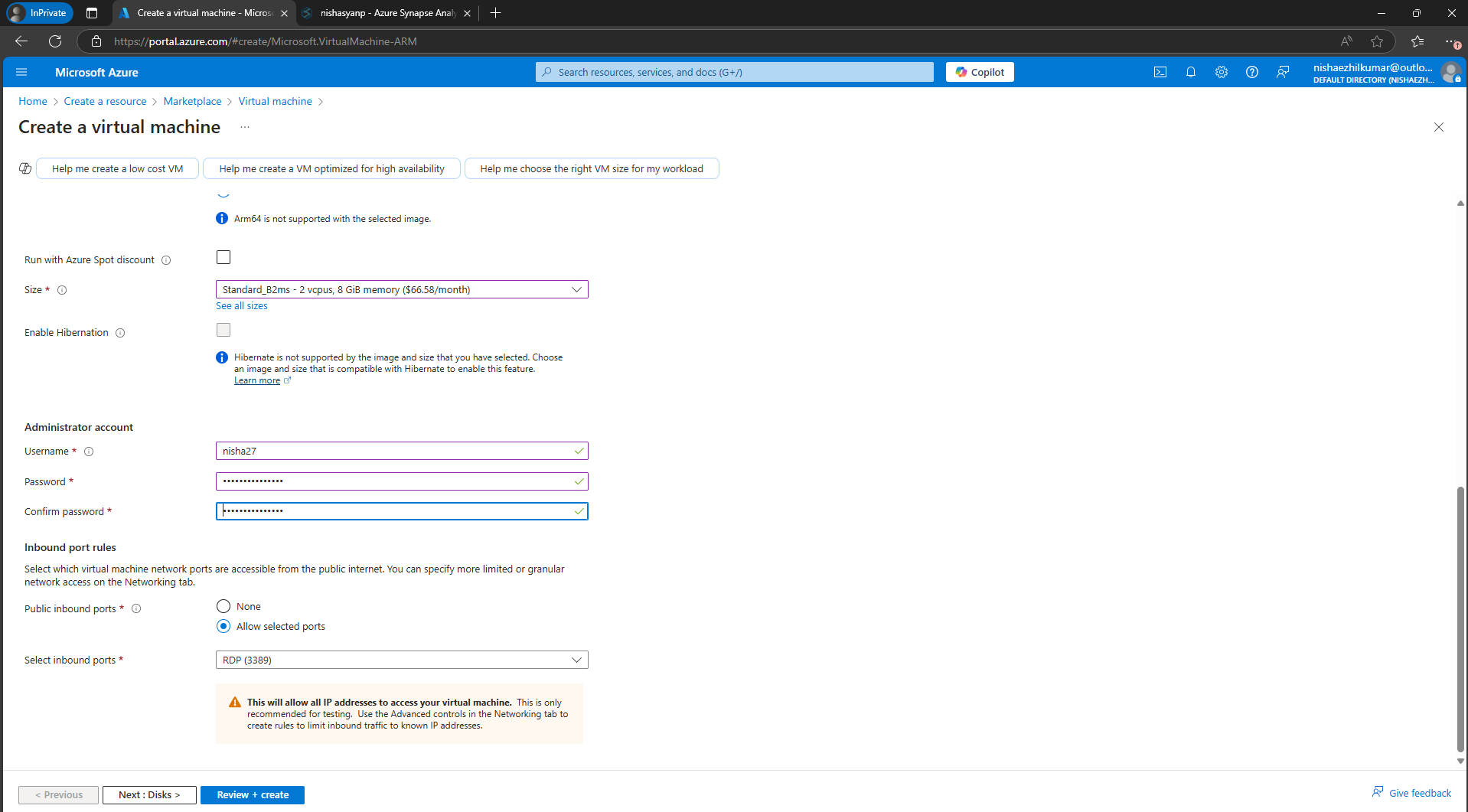
3. Creating a Virtual Machine (VM) with SQL Server

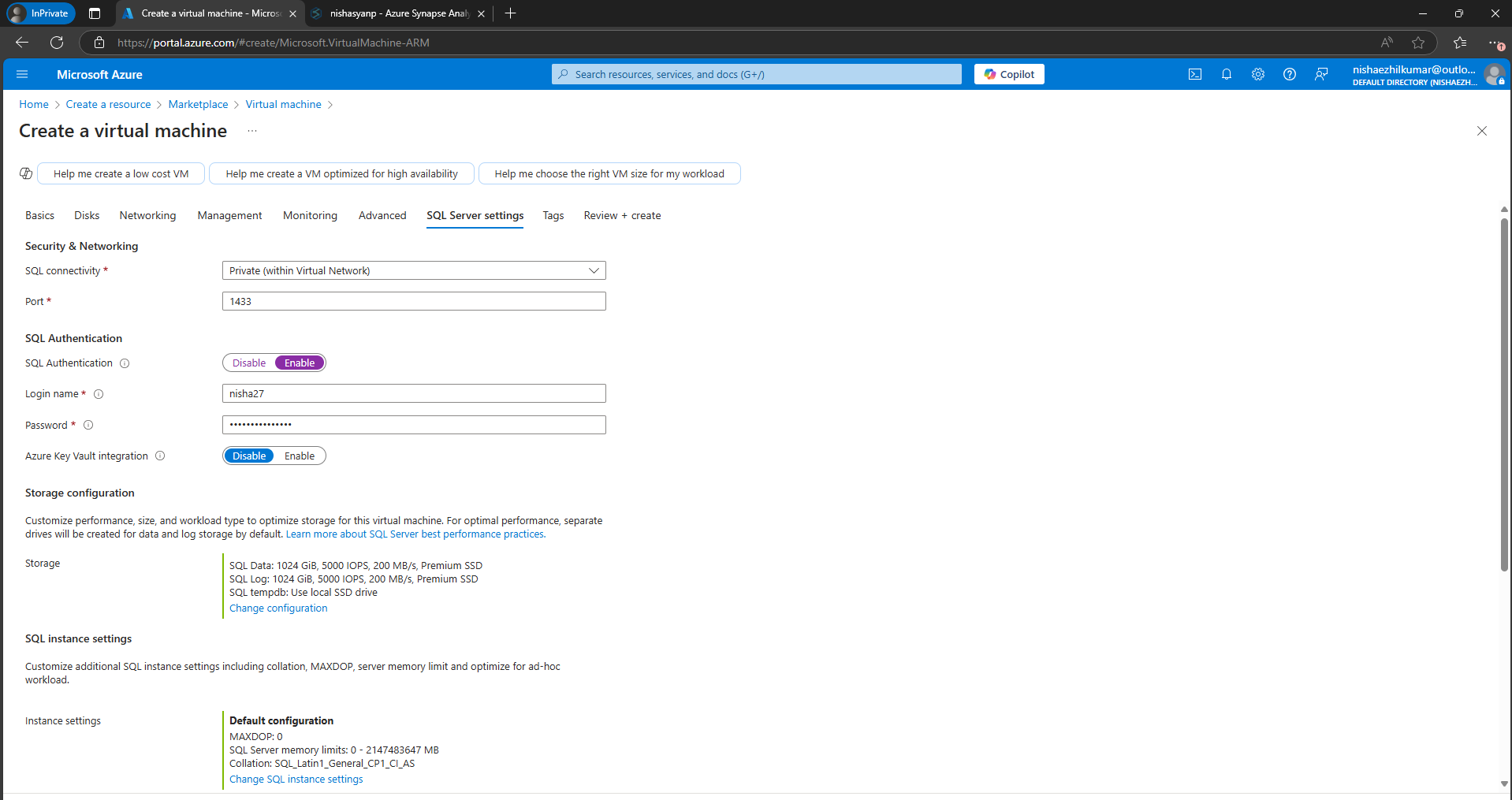
\* Create an Azure Virtual Machine (VM) and install SQL Server (On-Premises Setup).

\* Enable SQL Authentication during configuration.

\* Verify connectivity to the SQL Server instance.

\* Immediately shutdown the VM after creation to avoid unnecessary costs.





I am not sure about how to stop it, so I am stopping at this step